Think Big

Architecture is Recursive

Stefan Priebsch, thePHP.cc

IPC 2013, Munich

sharing experience


BANK
thePHP.cc

```
class BankAccount
{
    private $balance = 0;

    public function getBalance()
    {
        return $this->balance;
    }

    private function setBalance($balance)
    {
        if ($balance < 0) {
            throw new BankAccountException;
        }

        $this->balance = $balance;
    }

    ...
}
```

```php
class BankAccount
{
    public function depositMoney($amount)
    {
        $this->setBalance($this->getBalance() + $amount);

        return $this->getBalance();
    }

    public function withdrawMoney($amount)
    {
        $this->setBalance($this->getBalance() - $amount);

        return $this->getBalance();
    }

    ...
}
```

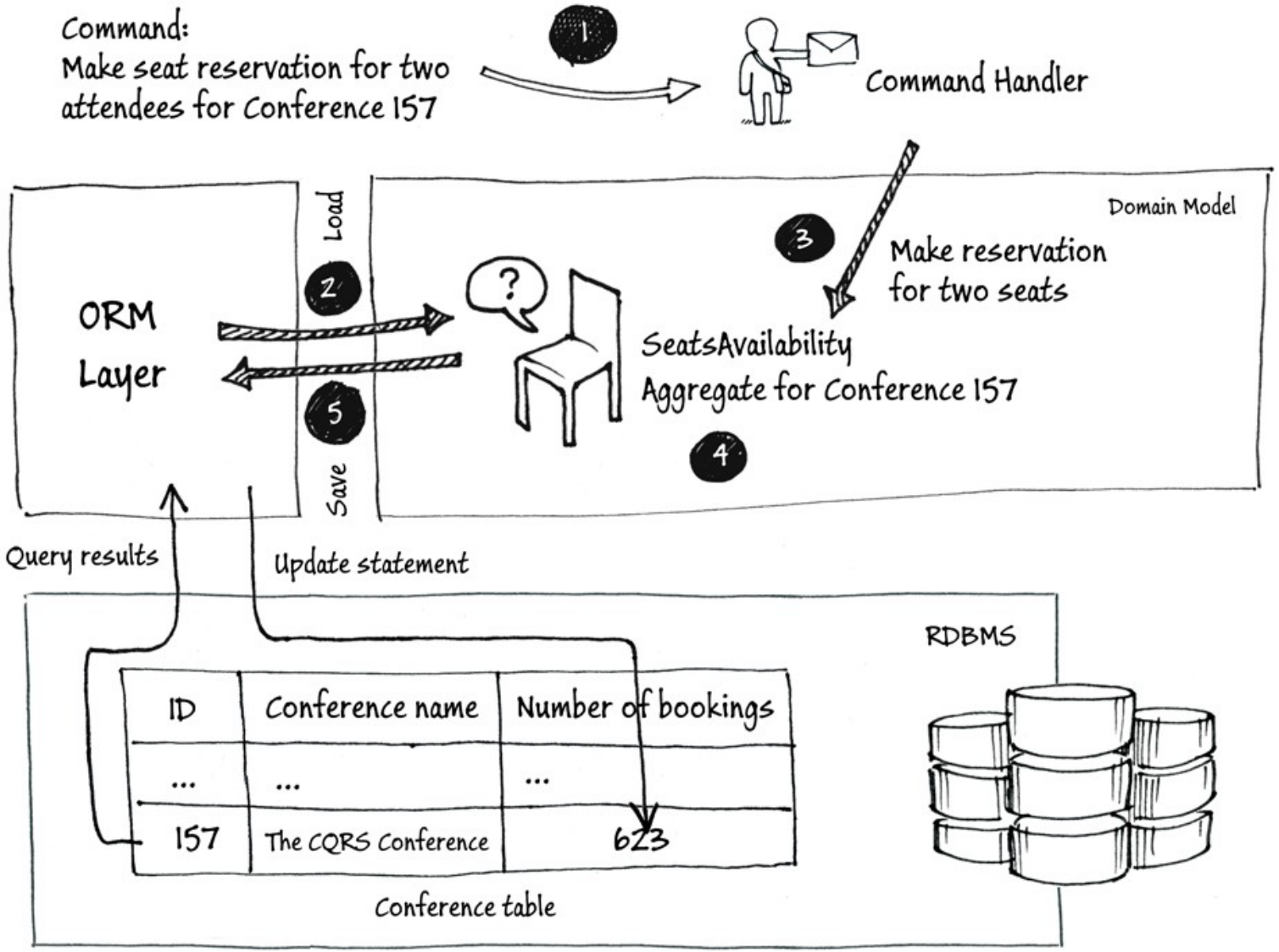

BANK
thePHP.cc

Goodbye, ACID

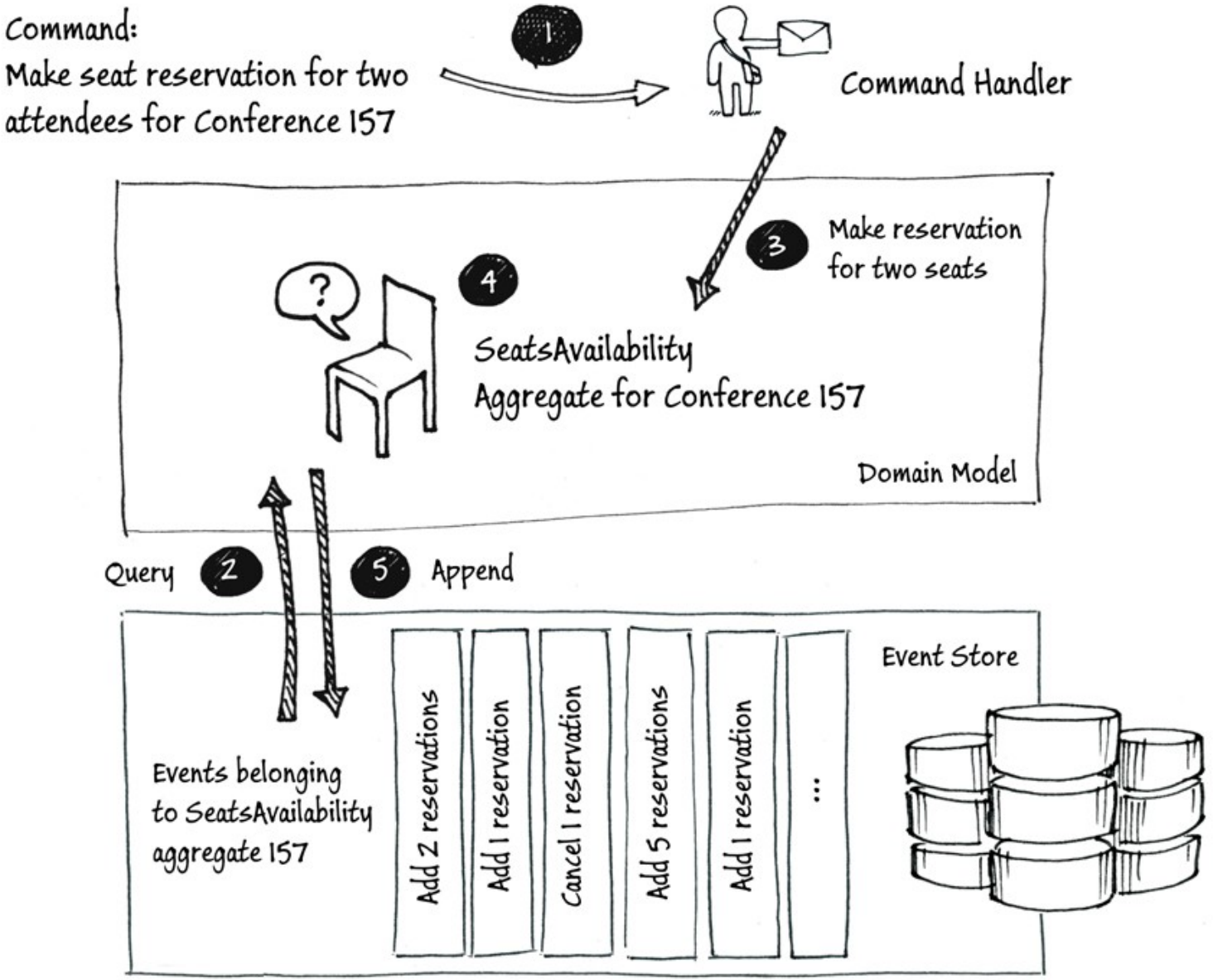
```php
class BankTransaction
{
    private $amount;

    public function __construct(
        $amount,
        BankAccount $from,
        BankAccount $to
    )
    {
        $from->withdraw($this);
        $to->deposit($this);
        $this->amount = $amount;
    }

    public function getAmount()
    {
        return $this->amount;
    }
}
```

```php
class BankAccount
{
    private $deposits = array();
    private $withdrawals = array();

    public function getBalance()
    {
        $balance = 0;

        foreach ($this->deposits as $deposit) {
            $balance += $deposit->getAmount();
        }

        foreach ($this->withdrawals as $withdrawals) {
            $balance -= $withdrawals->getAmount();
        }

        return $balance;
    }

    // ...
}
```

```php
class BankAccount
{
    private $deposits = array();
    private $withdrawals = array();

    public function deposit(BankTransaction $transaction)
    {
        $this->deposits[] = $transaction;
    }

    public function withdraw(BankTransaction $transaction)
    {
        $this->withdrawals[] = $transaction;
    }

    // ...
}
```



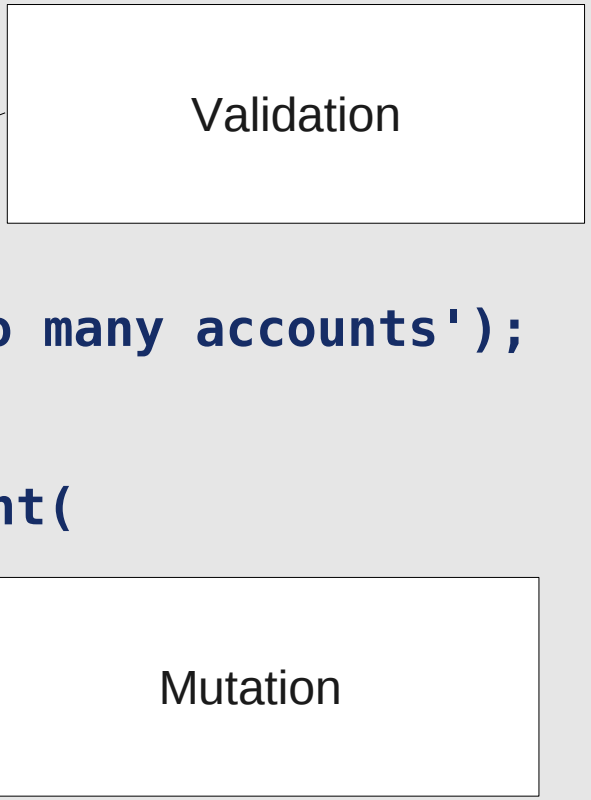
http://msdn.microsoft.com/en-us/library/jj591559.aspx



http://msdn.microsoft.com/en-us/library/jj591559.aspx

Event Sourcing

```
class Customer
{
    public function openAccount()
    {
        if (count($this->accounts) >= 5) {
            throw new CustomerException('Too many accounts');
        }

        $this->handle(new AccountCreatedEvent(
            new Uuid(),
            $this->getId()
        ));
    }

    ...
}
```



Separate validation and mutation

```
class Customer
{
    ...

    private function handleAccountCreatedEvent($event)
    {
        $this->accounts[] = new Account($event->getAccountId());
    }
}
```

Recreate objects from Events

```
class Customer
{
    public function __construct(Uuid $id, array $events = array())
    {
        $this->id = $id;
        $this->events = $events;

        if (count($events) > 0)  {
            $this->replayEvents($events);
        }
    }

    ...
}
```

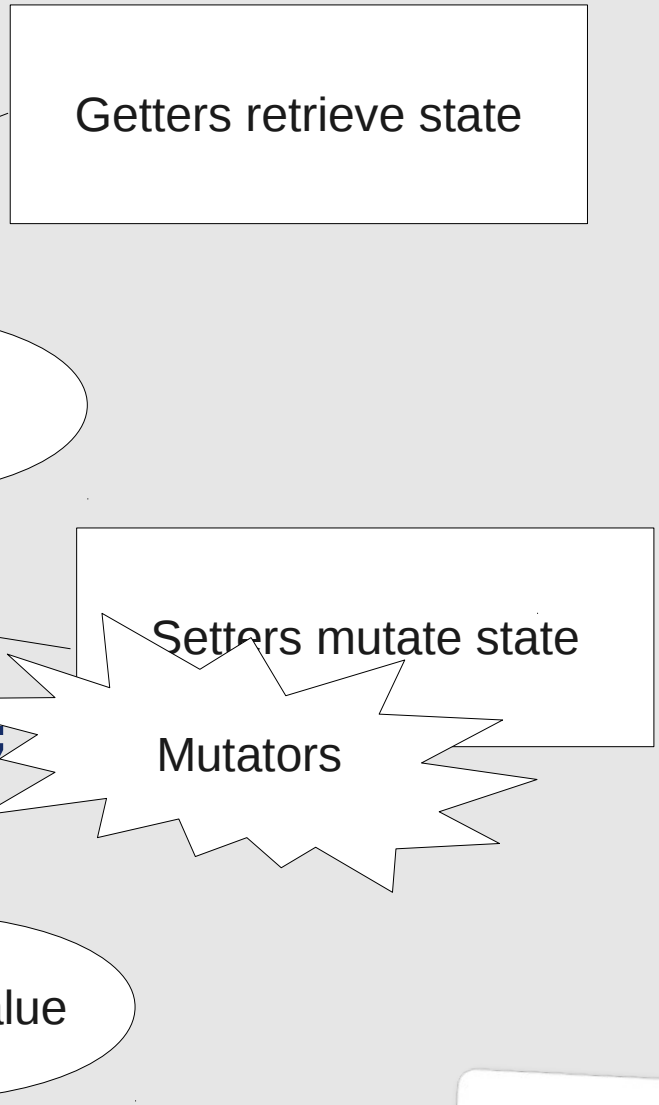
```php
class CreateAccountCommand
{
    private $userId;
    private $description;

    public function __construct($userId, $description)
    {
        $this->userId = $userId;
        $this->description = $description;
    }

    public function getUserId()
    {
        return $this->userId;
    }

    public function getDescription()
    {
        return $this->description;
    }
}
```

```php
class AccountCreatedEvent
{
    private $parent;
    private $userId;
    private $description;

    public function __construct(
        UUID $parent,
        $userId,
        $description
    ) {
        $this->parent = $parent;
        $this->userId = $userId;
        $this->description = $descriptions;
    }

    // ...
}
```

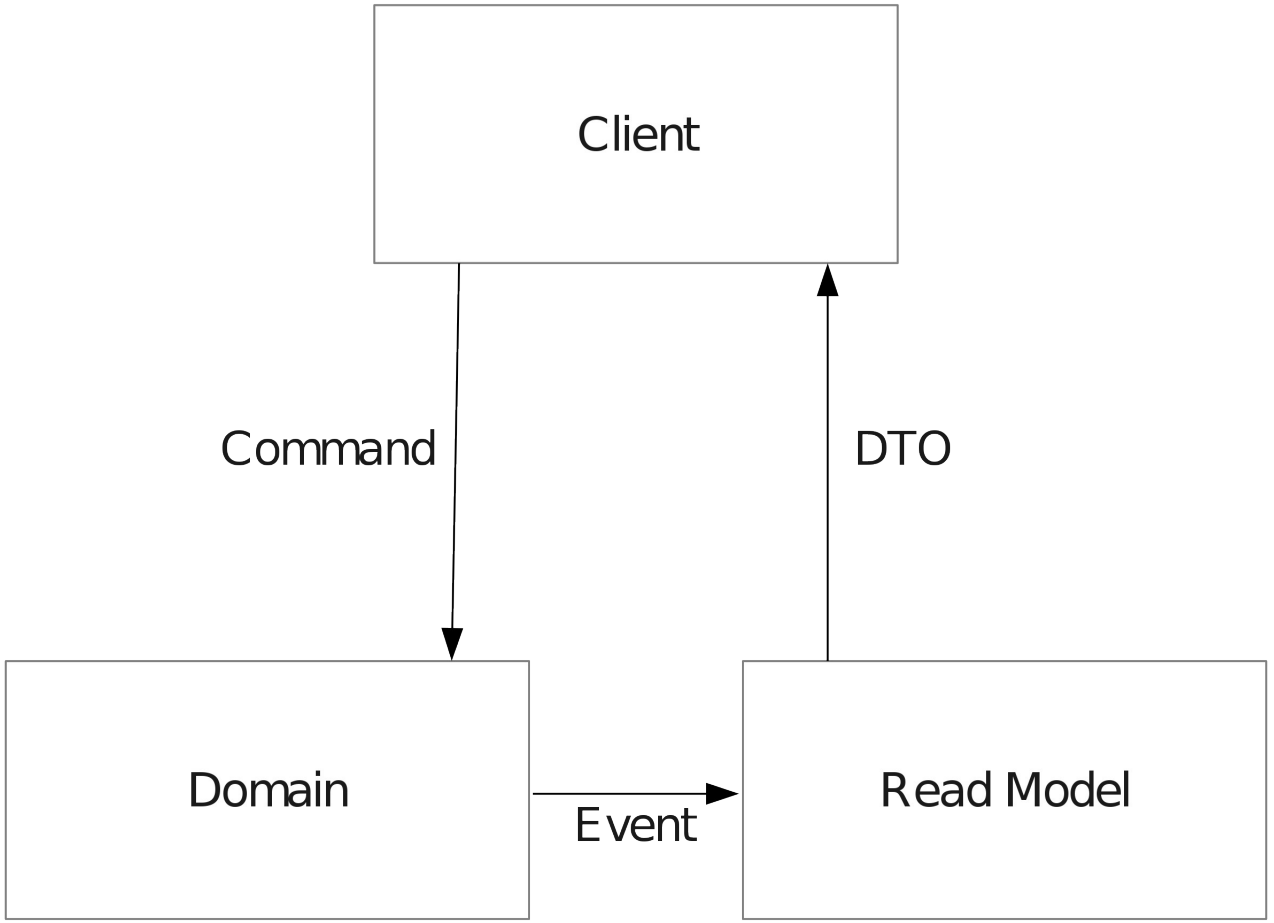
```
class AccountCreatedEvent
{
    // ...

    public function getParent()
    {
        return $this->parent;
    }

    public function getUserId()
    {
        return $this->userId;
    }

    public function getDescription()
    {
        return $this->description;
    }
}
```

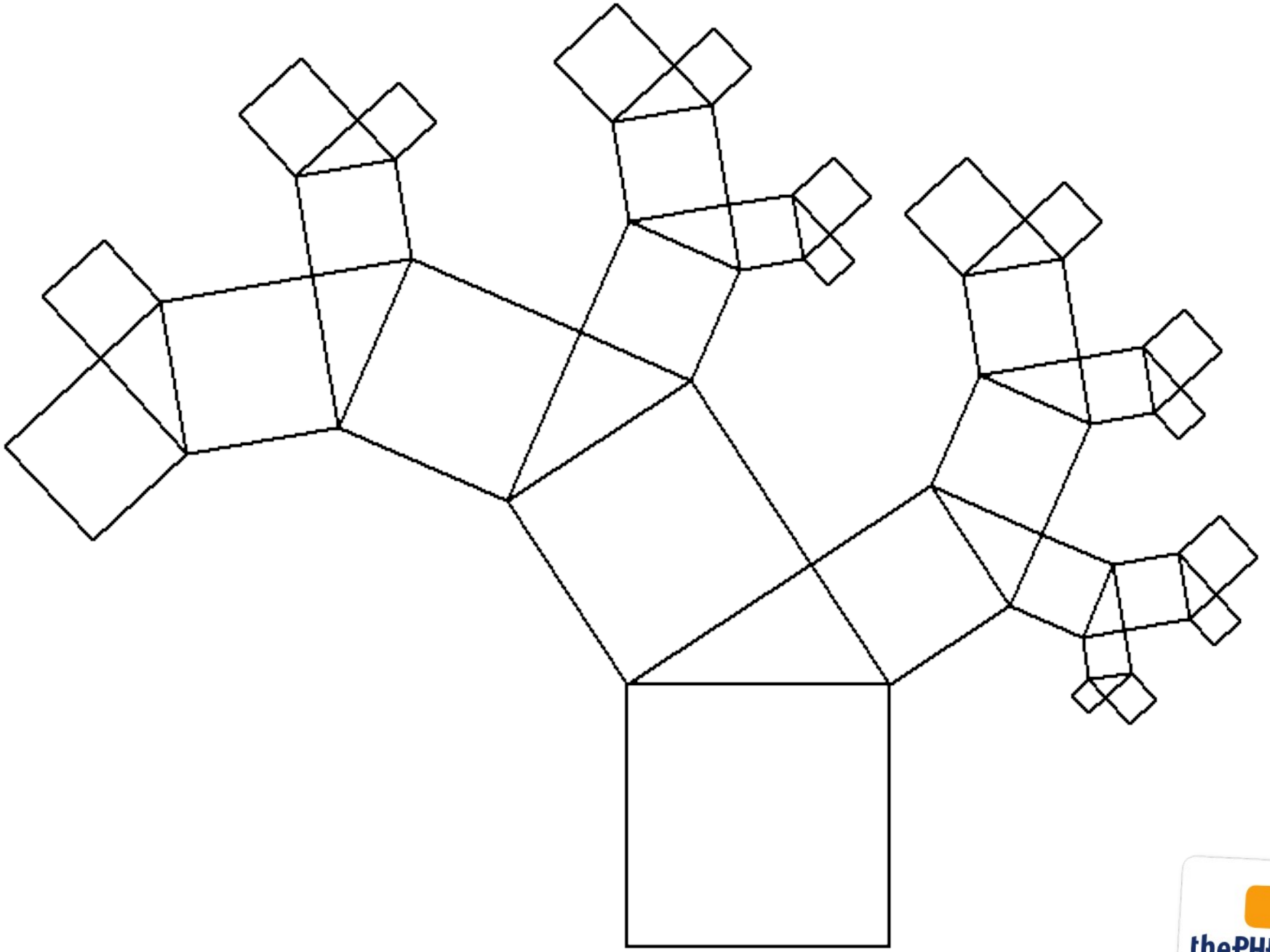
Command-Query Responsibility Segregation

```php
class BankAccount
{
    private $balance = 0;

    public function getBalance()
    {
        return $this->balance;
    }

    private function setBalance($balance)
    {
        if ($balance < 0) {
            throw new BankAccountException;
        }

        $this->balance = $balance;
    }

    ...
}
```



CQRS and Event Sourcing


Client
Command
DTO
Domain
Event
Read Model


thePHP.cc

Architecture is Recursive

http://priebsch.de

http://thePHP.cc
http://talks.thePHP.cc
stefan@thephp.cc

@spriebsch

sharing experience

sharing experience